

Human Pose Estimation using Machine Learning

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

with

TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Pushpendra Singh, pushpendrasingh10720@gmail.com

Under the Guidance of

P.Raja, Master Trainer, Edunet Foundation

ACKNOWLEDGEMENT

I would like to express my gratitude to all the individuals who guided and supported me during this project.

Firstly, I would like to thank my supervisor, **Mr. P Raja**, for his exceptional guidance, constructive feedback and consistent encouragement throughout the duration of this project. I am profoundly grateful to **Mr. P Raja** for giving me a chance for my practical knowledge exposure.

I also thankful to “**Microsoft-SAP**” giving me an opportunity to launch my career in this challenging area, I really fortunate to work under the guidance of who despite his busy schedule helped me to upgrade my knowledge base.

I also wish to acknowledge the “**TechSaksham**” initiative by “**Microsoft & SAP**” for providing this transformative learning opportunity. Finally we would like to express our gratitude towards family, peers and friends for their kind cooperation and encouragement which helped us in completion this project.

PUSHPENDRA SINGH

ABSTRACT

This report focuses on the project made, titled **Human Pose Estimation using Machine Learning**, HPE-Human Pose Estimation is the task that aims to predict the location of human joints from the images and videos. This task is used in many applications, such as sports analysis and surveillance system. HPE model is difficult many challenges like crowded scenes and occlusion, must be handled. Human Pose Estimation is becoming a popular field of research in the last two decades.

Understanding Human behavior in images gives useful information for a large number of computer vision problems and has many applications like scene recognition and pose estimation using machine learning. In this project we proposed approach for human activity recognition and classification using a person's pose skeleton in images. This project divided into two parts; a single person pose estimation and activity classification using pose.

Pose estimation consist of 18 body key points and joints locations. We have used OpenPose library for pose estimation and we have prepared our dataset, divided into two parts, one is used to train the model, and another is used to validate our proposed model's performance.

The methodology of human pose estimation is mainly used for the purpose of training the robots to incorporate in a way which actions are performed in reality. The human pose estimation is the highly exploring in the field of computer vision research. The main objective of dynamic pose estimation is to estimate the human pose in all the available dataset. It begins with mapping the skeletal coordinates are being obtained the coordinates. The features extraction has taken place separately for RGB and depth dataset.

The computed data is being present in the same format as that of the vector space that is being allocated for the feature extractor. The classifiers such as support vector machine (SVM), k-nearest neighbor (KNN), and the decision tree are being used. The major use of this learning process is to help the robot to train it similar to that of the human functionalities. These functionalities can be used in any learning and training procedures of how every action take place is leaned in this process and thereby training any system with similar measures is simplified with this procedure.

The project conclude human pose estimation using machine learning represents a rapidly evolving field with vast potential across diverse applications from enhancing fitness and rehabilitation to enabling immersive experiences in augmented and virtual reality, the implications of accurate pose detection are profound ad advancements in algorithms and computing power continue , we can expect improvements with other modalities, real-time processing capabilities, and integration with other modalities, leading to more intuitive and responsive systems.



TABLE OF CONTENT

Abstract	I
Chapter 1. Introduction	1
1.1 Problem Statement	
1.2 Motivation	
1.3 Objectives	
1.4 Scope of the Project	
Chapter 2. Literature Survey	5
2.1 Review relevant literature or previous work in this domain	
2.2 Mention any existing models, techniques, or methodologies	
2.3 Pose Track	
Chapter 3. Proposed Methodology	8
3.1 System Design	
3.2 Implemented Human Pose Estimation Code	
3.3 Train Mediapipe feed with the help of web	
Chapter 4. Implementation and Results	13
4.1 Preparing Data set model for pose estimation using ML	
4.2 Trained pose estimation model with the help of ML	
4.3 Result-1 and Result-2	
4.4 GitHub Link for Code	
Chapter 5. Discussion and Conclusion	19
References	21

LIST OF FIGURES

Figure No.	Figure Caption	Page No.
Figure 1	Motivation	I
Figure 2	Determining Human Body Joints	V
Figure 3	Pose Tracking	VII
Figure 4	System Design	VII
Figure 5	Preparing Data Set for pose estimation	IX
Figure 6	Implement Mediapipe Feed Through Web	X
Figure 7	Make Proper Pose Detection	XIII
Figure 8	Trained Data Set for pose estimation	IV
Figure 9	Snap Shot of Result-1	XVI
Figure 10	Snap Shot of Result-2	XVII

CHAPTER 1

Introduction:

Human pose estimation is mainly used for the purpose of training the robots to incorporate in a way which the action are performed in reality. The human pose estimation is highly exploring field in computer vision. The objective of dynamic pose estimation the human pose in all the available dataset. The features extraction has taken place separately for RGB and depth dataset. The major use of this learning process is to help the robot to train it similar to that of human functionalities. The proper understanding of how every action takes place is learned in this process and thereby training any system with similar measures is simplified with this procedure.

1.1 Problem Statement:

Understanding human movements and body postures is challenging, especially in areas like sports, healthcare and surveillance. Without Human Pose Estimation, tasks like motion analysis and injury prevention become manual, slow and prone to mistakes.

1.2 Motivation:

Pose estimation is a computer vision technique for tracking the movements of a person or an object. It is usually performed by finding the location of key points for the given objects. We can compare various movements and postures based on these key points and draw insights pose estimation is used in augmented reality, animation, gaming, and robotics.

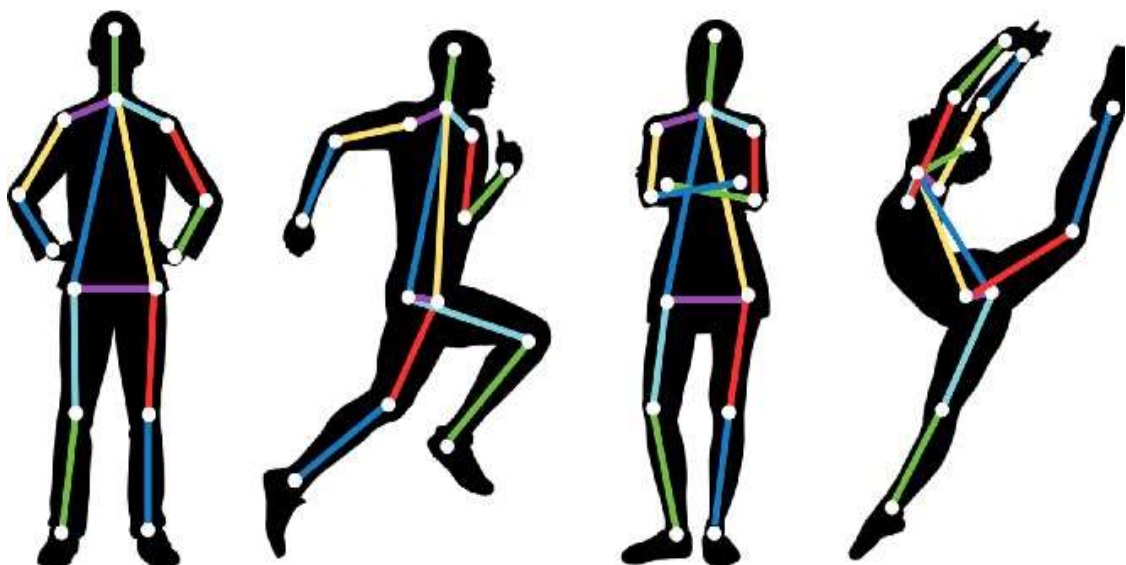


Fig.01

Pose estimation is a computer vision technique to track the movements of a person or an object. This is usually performed by finding the location of key points for given object. Based on these key points we can compare various movements and postures and draw insights.

1.3Objective:

The main objective of Pose estimation is actively used in the field of augmented reality, animation, gaming, and robotics. There are several models present today to perform pose estimation. Some of the methods for pose estimation are given below.

- Open pose
- Pose net
- Blaze pose
- Deep pose
- Dense pose
- Deep cut
- Tensorflow
- Regional Multi-person Pose Estimation
- OpenPifPaf
- YoloV8

Choosing any one model over another may totally depend upon the application. Also the factors like running time, size of the model, and ease of implementation can be various reasons to choose a specific model.

What are the potential applications and the impact:

- Movies
- Virtual Reality
- Animation
- Pose-Based Games
- Mental Development
- Sports Action Analysis
- Surveillance
- Physiotherapy
- HCI- Human Computer Interaction

1.AI Fitness and training applications:

Human pose estimation got the most attention in the context of AI fitness applications, as it can be applied to analyze movements of athletes in different scenarios using just a smartphone camera.

HPE- based fitness apps can be generally split into two categories.

1. **Sports performance analytics:** Those applications provide athletes with insights on how they perform a certain movements over a period of time, and can show accurate metrics for exercises. These can be height of a hip in a jump, lever angle in power movements, changes in technique between repetitions etc.
2. **AI coaching and corrective feedback:** This category is meant to show whether a user is performing the exercise correctly technique-wise. Such hints might include posture correction, biomechanic tips, and overall mentoring through comparative training.
3. **Animation and gaming applications:** Game development is a tough industry with a lot of complex tasks that require knowledge of human body mechanics. Body pose estimation is widely used in animation of game character to simplify this process by transferring tracked key points in a certain position to the animated model. This process of work resembles motion tracking technology used in video production, but it does not require a large number of sensors placed on the model. Instead, we can use multiple cameras to detect the motion pattern and recognize it automatically. The data fetched that can be transformed and transferred to the actual 3D model in the game engine.

1.4 Scope of the Project:

- Developing models that can accurately estimate pose in challenging conditions (eg..occlusions, complex backgrounds, diverse body types.)
- Optimizing algorithms for deployment on mobile devices and IoT platforms, enabling real-time pose estimation without relying on cloud computing.
- Combining pose estimation with other data types, such as audio, or environments sensors to create more comprehensive models for applications like smart home system and healthcare monitoring.
- Utilizing advancements in deep learning such as transformer models for more effective pose estimation.

- Applying pose estimation for real-time behavioral analysis in various settings from retail to healthcare to derive actionable insights.

Limitations:

- Occlusions: when parts of body are obscured by other objects or body parts, pose estimation accuracy decreases.
- Complex Poses: Unusual or complex can be challenging for models to accurately predict.
- Variability in Lighting and Backgrounds: Changes in lighting conditions or complex backgrounds can interfere with accurate pose estimation.
- Real-Time Processing: Real-time pose estimation require substantial computational resources, which may not always be available.
- Generalization: Models trained on specific dataset may not generalize well to different populations, environments, or clothing variations.
- Multi-Person Scenarios: Accurately estimating poses in scenes with multiple people interacting can be challenging.

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

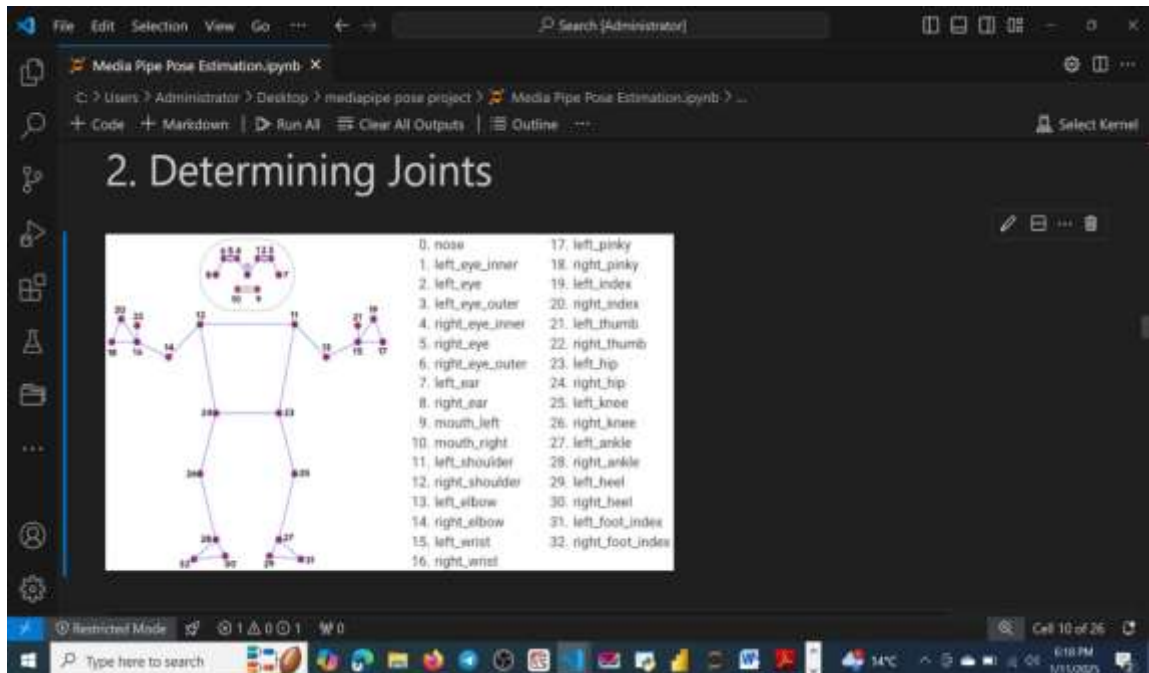


Fig.02

Human pose *estimation* problem has been heavily studied in computer vision. It has important applications in human-computer interaction, virtual reality and action recognition. Existing research works falls into two categories 2D or 3-D pose estimation. State of the art techniques are able to achieve accurate predictions across a wide range of settings. We focus on previous work on pose estimation a popular approach is to train a neural network learning model to directly regress joints location.

Pose *estimation* benchmark dataset, the resulting networks do not generalize to image in the wild due to the specific capture environments utilized these benchmark dataset. A standard approach to address the domain difference between 3D human pose estimation datasets and images in the wild id to split the task into two separate sub tasks. The sub-task estimate 2D joints locations. The sub task can utilize any existing 2D human pose estimation method and can be trained from dataset of in the wild images. The seconds subtask regress the 3D locations of these 2D joints.

2.2Mention any existing models, techniques, or methodologies related to the problem.

Data Planning:

1. Define the problem and objective: Estimate human poses from image to video streams.
2. Set Success Metrics: KPI include detection accuracy, speed, and robustness against occlusions.
3. Gather Requirements: Use libraries- Opencv, Streamlit, Numpy, Matplotlib and Pillow.
4. Create Timeline: Divide tasks- preprocessing, model training, visualization, deployment.

Data Acquisition and Preprocessing:

1. Collect Data: Use dataset like coco or MPII or capture human pose images and videos.
2. Clean Data: Filter out incomplete or mislabeled data, ensure uniform keypoint labeling.
3. Transform Data: Preprocess image with Opencv resize, normalize, and annotate poses.
4. Feature Engineering & Data Split: Extract pose coordinates split into training, validation and test sets.

Machine Learning model require a large amounts of data to perform specific tasks accurately. Human pose estimation models in a particular require diverse data to handle challenges such as varying backgrounds, illumination and clothing. Fortunately existing dataset address these challenges by offering a diverse recording environment. Some dataset also provide different pose activities that contain complex poses and occlusion problems.

Dataset play a crucial role in training and testing human pose estimation using machine learning model. Human pose estimation dataset typically consist of images, video, or both that capture human subjects in various pose ID, Joints visibility and activity name.

These annotations helps address HPE challenges such as occlusion, tracking multiple poses and handling complex poses.

The most well know dataset used for 2D pose estimation that are available for access are LSP, FLIC, MPII, COCO, CrowdPose. These datasets are commonly used for estimating single/ multiple poses in images while PenAction, JHMDB and PoseTrack are used to estimate poses from videos.

Pose Track:

The pose track dataset I widely used to train models for estimating and tracking multi person poses. This dataset contains challenging scenarios involving highly occluded individuals crowded environments with complex movements.

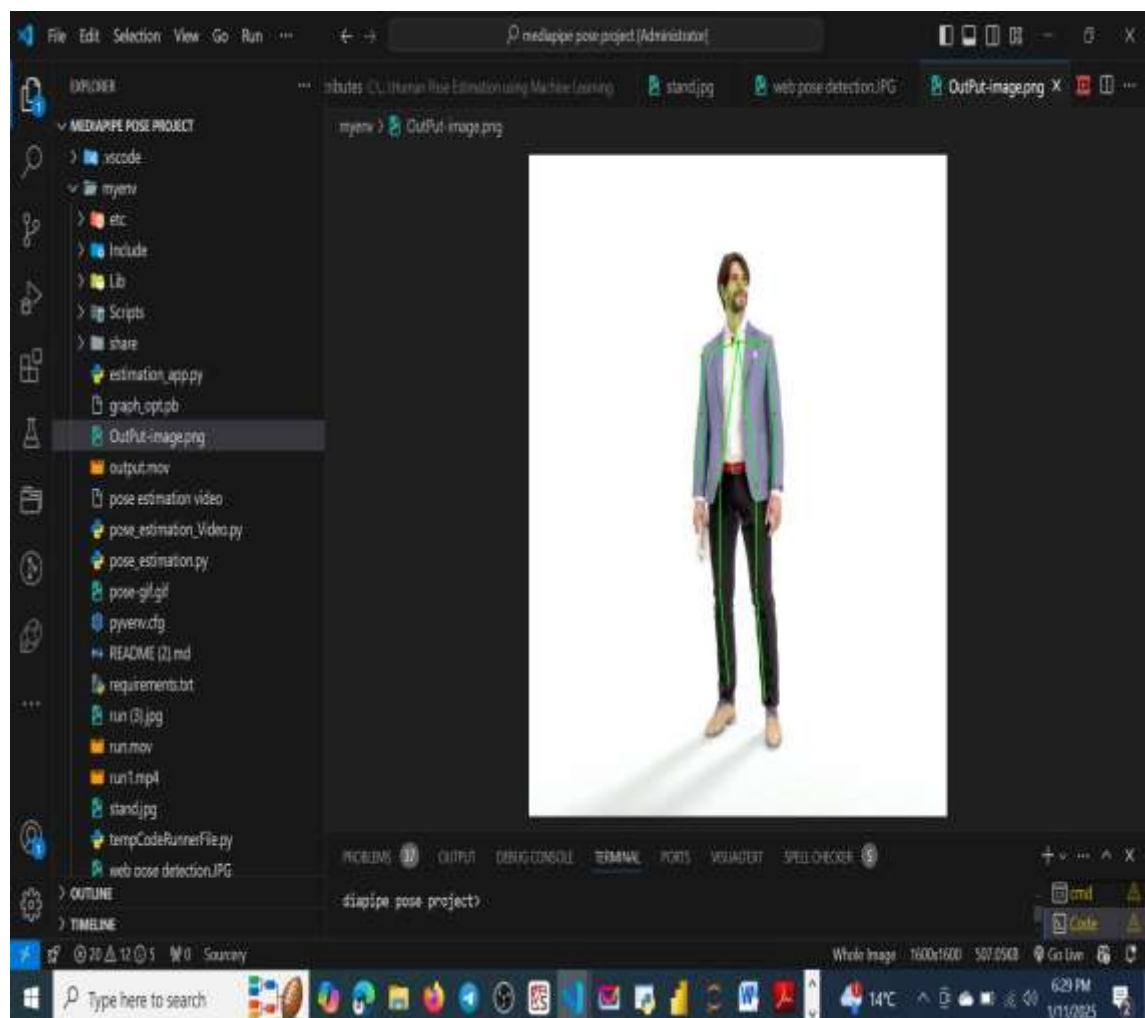


Fig.03

CHAPTER 3

Proposed Methodology

3.1 System Design

If there is only one person in images and two approaches can be used to estimate the pose: regression based and detection based. Regression based methods use an end to end framework to learn a mapping from in an image to the joint coordinates of the body directly producing joints coordinate. This system is designed represent the pipeline of the regression and detection approaches. Both approaches have their pros and cons. While detection learning is supervised by dense pixel information, direct regression learning of a single point is challenging sue to being a highly nonlinear problem.

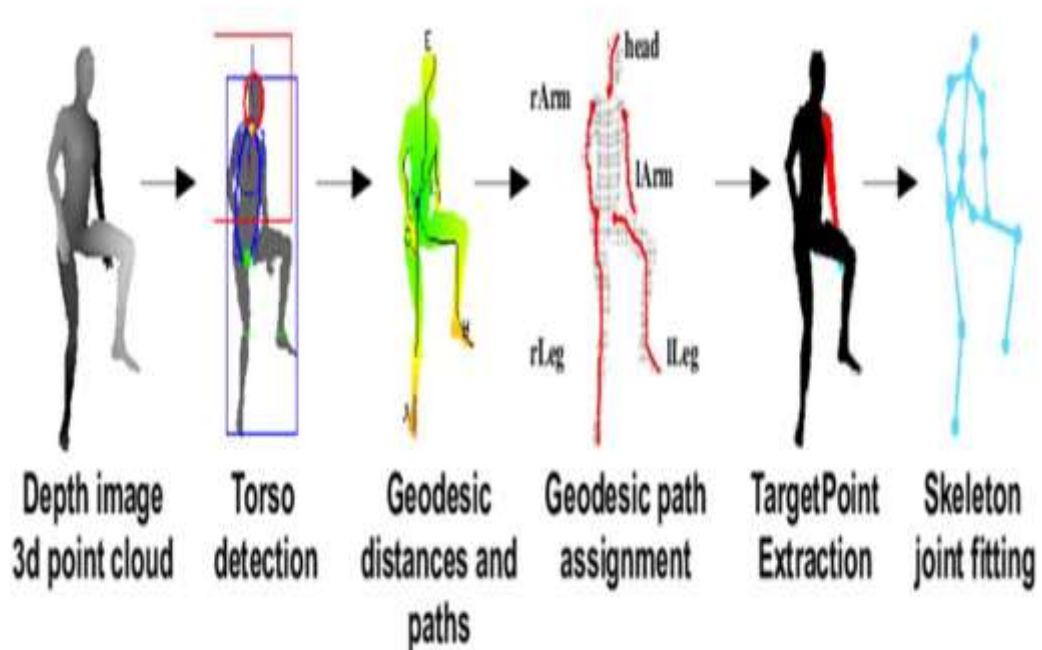


Fig.04



Implemented Pose Estimation Code:

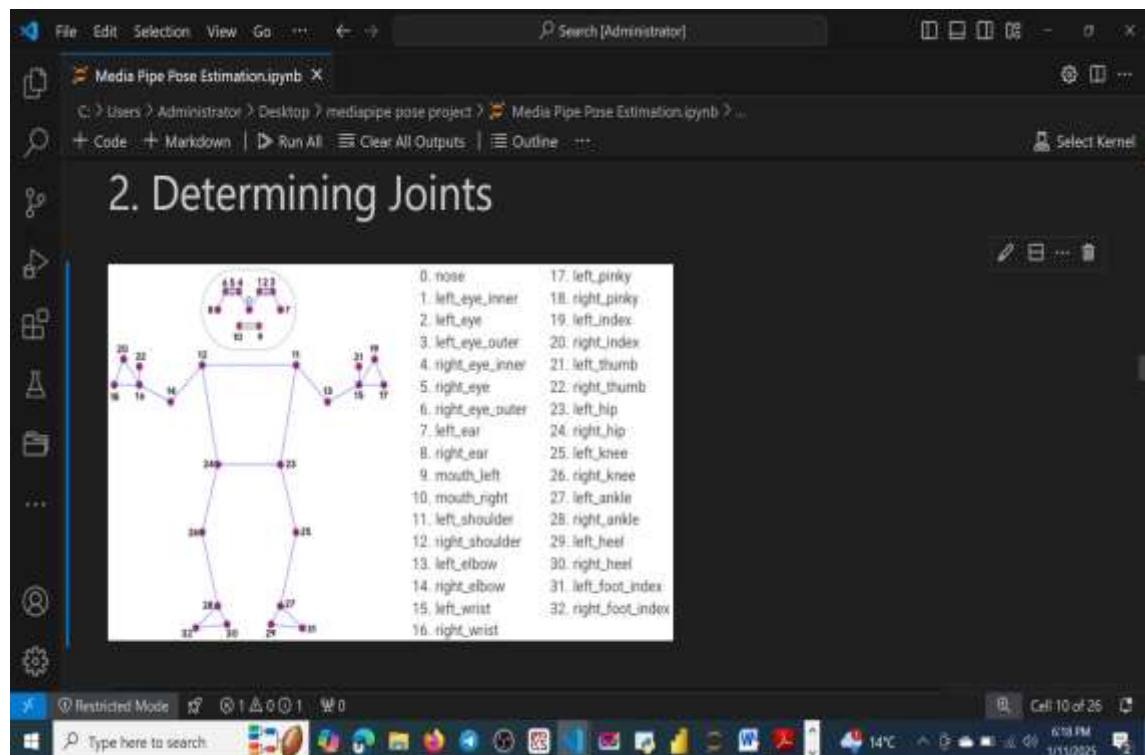


Fig.05

```

import cv2
import numpy as np
import matplotlib.pyplot as plt

BODY_PARTS = { "Nose": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3,
               "RWrist": 4,
               "LShoulder": 5, "LElbow": 6, "LWrist": 7, "RHip": 8,
               "RKnee": 9,
               "RAnkle": 10, "LHip": 11, "LKnee": 12, "LAnkle": 13,
               "REye": 14,
               "LEye": 15, "REar": 16, "LEar": 17, "Background": 18 }

POSE_PAIRS = [ ["Neck", "RShoulder"], ["Neck", "LShoulder"], ["RShoulder",
               "RElbow"],
               ["RElbow", "RWrist"], ["LShoulder", "LElbow"], ["LElbow",
               "LWrist"],
               ["Neck", "RHip"], ["RHip", "RKnee"], ["RKnee", "RAnkle"],
               ["Neck", "LHip"],
               ["LHip", "LKnee"], ["LKnee", "LAnkle"], ["Neck", "Nose"],
               ["Nose", "REye"],
               ["REye", "REar"], ["Nose", "LEye"], ["LEye", "LEar"] ]

```



```
width = 368
height = 368
inWidth = width
inHeight = height

net = cv2.dnn.readNetFromTensorflow("graph_opt.pb")
```

Train Pose estimation through MediaPipe feed using Machine Learning:

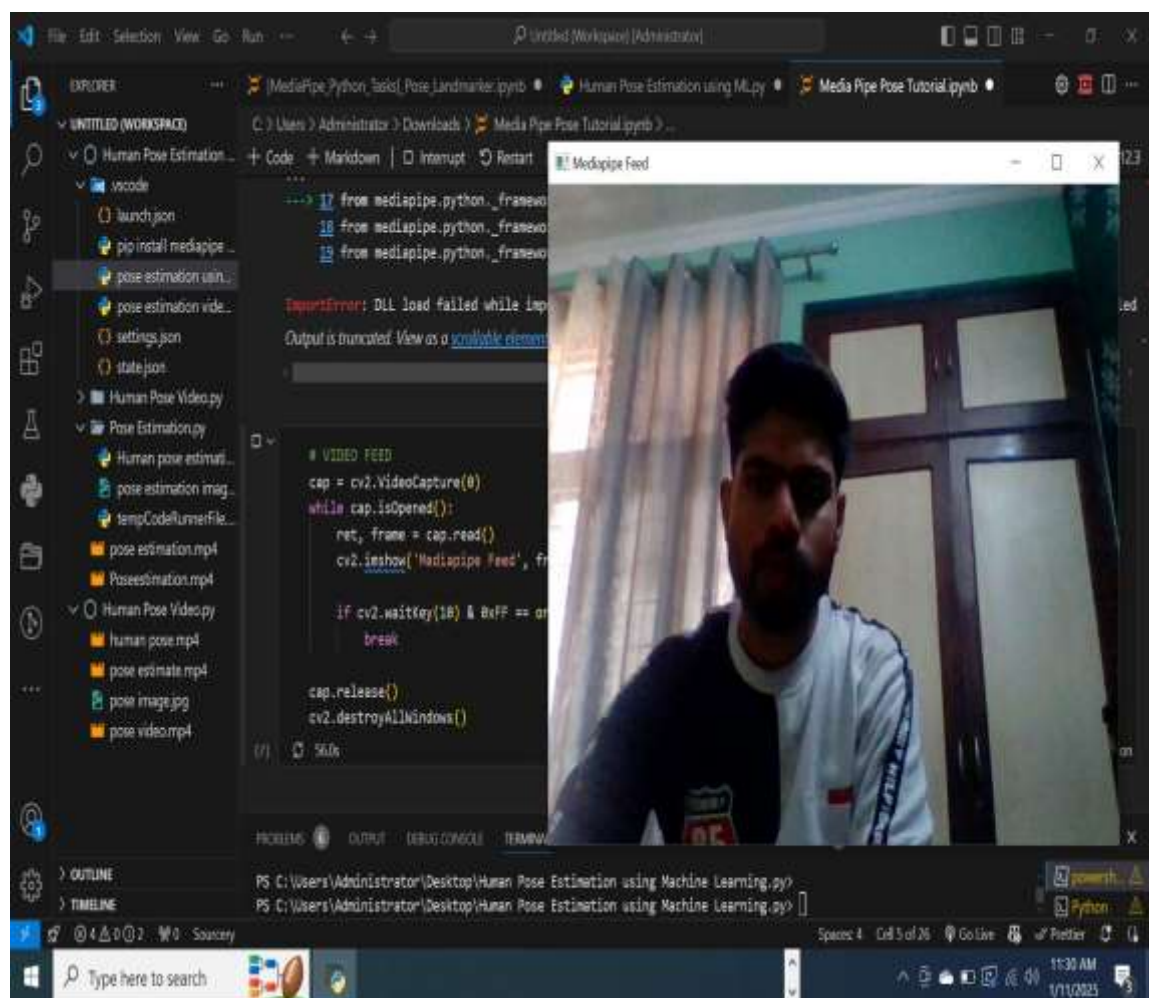


Fig.06


```
File Edit Selection View Go ... Search [Administrator]

MediaPipe Pose Estimation.ipynb X
C:\Users\Administrator\Desktop> mediapipe pose project > MediaPipe Pose Estimation.ipynb > ...
+ Code + Markdown ▶ Run All ⌵ Clear All Outputs ⌵ Outline — Select Kernel

1. Make Detections

cap = cv2.VideoCapture(0)
# Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, frame = cap.read()

        # Render image to RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False

        # Make detection
        results = pose.process(image)

        # Render back to BGR
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Render detections
        mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS,
                                  mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=2),
                                  mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
                                  )

        cv2.imshow('MediaPipe Feed', image)

        if cv2.waitKey(30) & key == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Fig.07

3.2 Requirement Specification

Software Requirement:

- Application : Anaconda
- Primary Language : Python
- Backend Technologies : Jupyter Notebook, VSCODE

Software Requirements Prerequisites:

- Streamlit
- Tensorflow
- Pytorch
- Opencv-python
- Pillow
- Numpy
- Pandas
- Mediapipe

Hardware Requirement:

- OS: Window
- Processor : intelcorei5 vpro8thGen
- Ram : 8GB
- Hard Drive ; 25GB and ABOVE

Code 2:

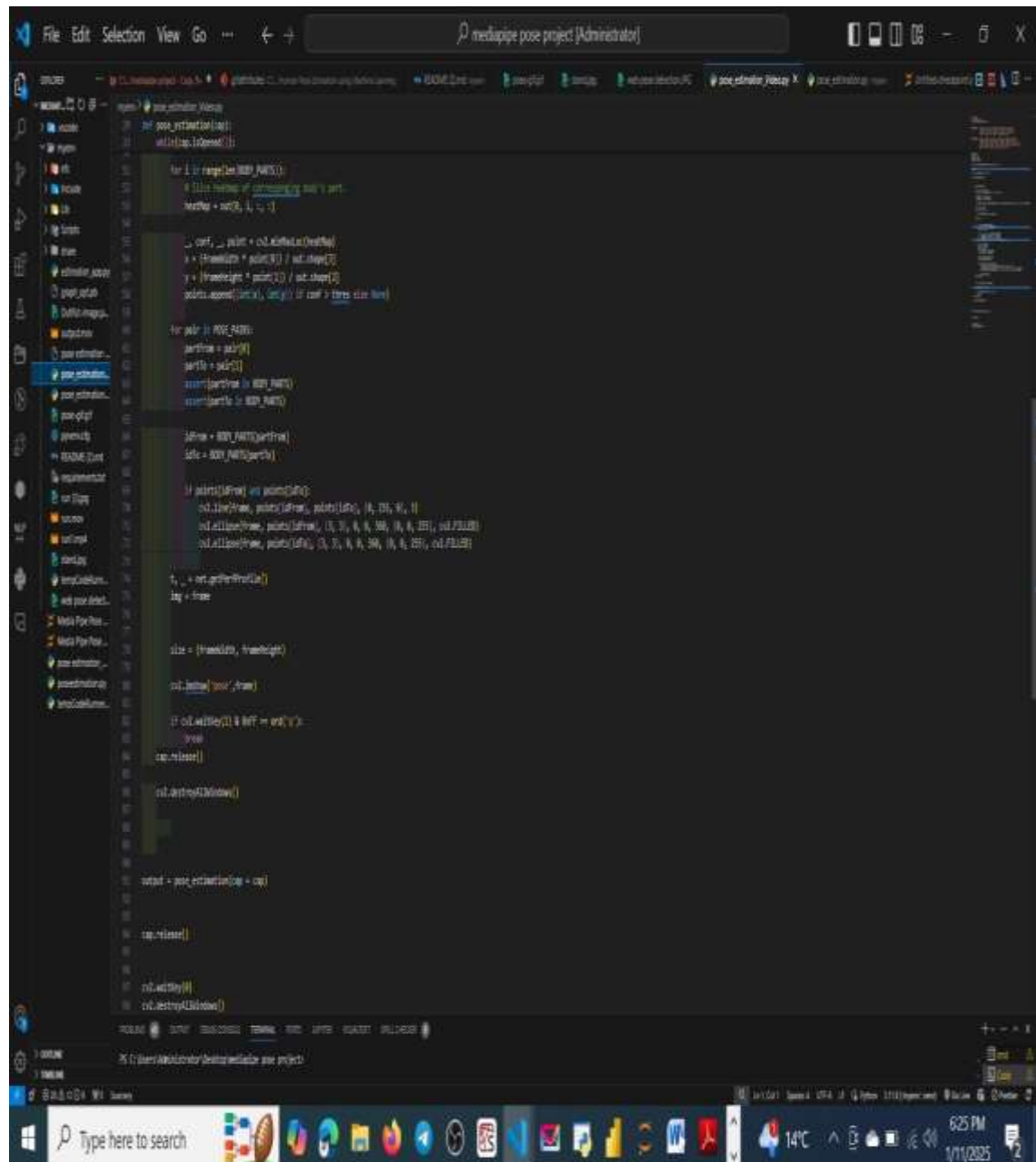
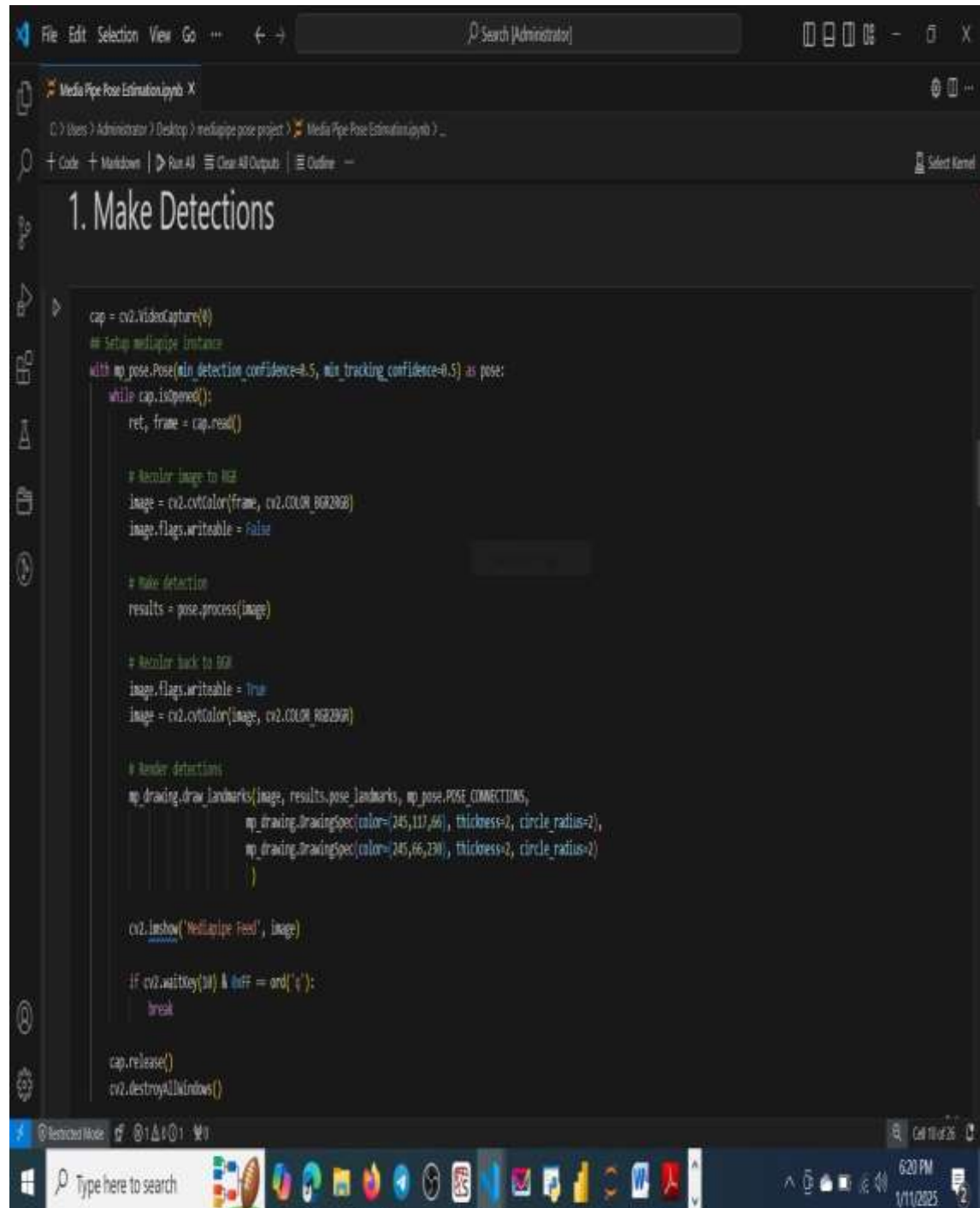


Fig. 09

Train Pose Estimation Model:



```
File Edit Selection View Go ... Search [Administrator]

Media Pipe Pose Estimation.ipynb X
C:\Users\Administrator\Desktop> mediapipe pose project > Media Pipe Pose Estimation.ipynb > ...
+ Code + Markdown ▶ Run All ⌵ Clear All Outputs ⌵ Outline — Select Kernel

1. Make Detections

cap = cv2.VideoCapture(0)
# Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
    while cap.isOpened():
        ret, frame = cap.read()

        # Render image to RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image.flags.writeable = False

        # Make detection
        results = pose.process(image)

        # Render back to BGR
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Render detections
        mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS,
                                  mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=2),
                                  mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
                                  )

        cv2.imshow("Mediapipe Feed", image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Fig.10

4.0 Snap Shots of Result:1

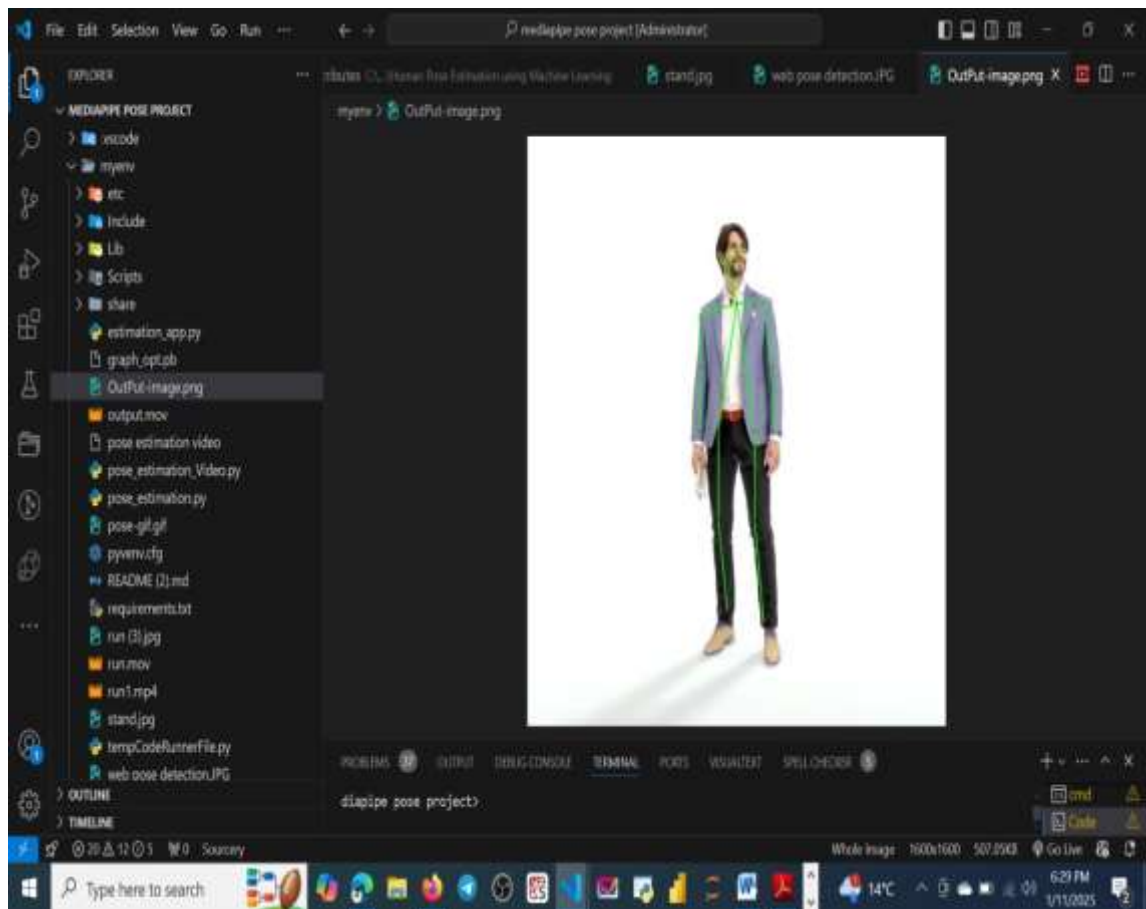


Fig.11

Explanation:

In the above snippet of code, we have first imported the necessary libraries that will help in creating the dataset. And each key points contain four attributes that are x and y coordinates of the z coordinates that represents landmark depth with hips as the origin and same scale as that of x and lastly the visibility score.

After the key points of all the images we have add a target value pose estimation that will act as a label for our machine learning model. The output contain both normalized coordinates (landmarks) and for each landmark And it represent visibility: the likelihood of the landmark being visible within image pose estimation using machine learning trained model.



4.1 Snap Shots of Result: 2

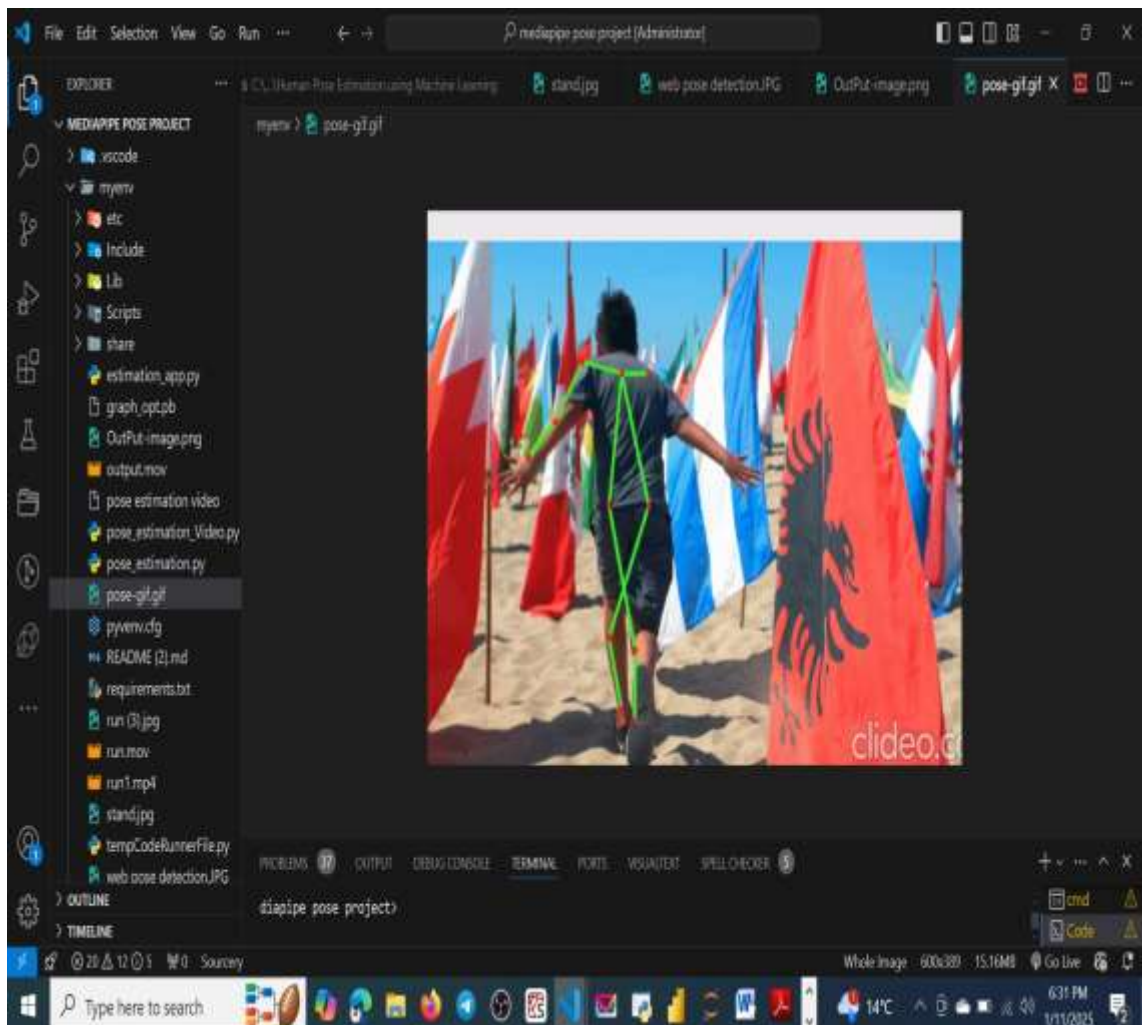


Fig.12

Explanation:

The Output contains the following normalized coordinates (Landmarks)

- **X and Y:** Landmark coordinates normalized between 0.0 and 1.0 by the image width (x) and height (y).
- **Z:** The landmark depth, with the depth at the midpoint of the body gesture.
- **Visibility:** The likelihood of the landmark being visible within the image.

4.1 GitHub Link for Code:

<https://github.com/pushpendra10720/Human-Pose-Estimation-using-Machine-Learning-project-.git>

4.2 URL of LinkedIn:

<https://www.linkedin.com/in/pushpendra-singh-8b61b325b/>

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

- Despite significant progress in estimating human pose from images or videos, existing machine learning models still face challenges with accuracy and efficiency, particularly in the presence of occlusions and crowded scenes. While many machine learning model commonly used in human pose estimation for their effectiveness in implicit features from images have explored CNN models.
- Improving the efficiency of human pose estimation tasks is not limited to enhancing models dataset labels also play a significant role. Only a few dataset provide additional visibility of body joints that can help address the challenge of occlusions. As occlusion is one the main challenges in 2D human pose estimation and researcher need to increase the number of occluded labels in datasets. Unsupervised / semi supervised and data augmentation methods are currently used address this limitation.
- Mediapipe and Streamlit and Opencv , yolo and OpenPose libraries used for detecting and estimating pose to identify suspicious behavior during the training dataset for human pose estimation in real-time.
- Developing model is that can accurately estimate poses in challenging conditions such as occlusions, complex backgrounds, diverse body types.
- Optimizing algorithms for deployment on mobile device and IoT platforms. Enabling real-time pose estimation without relying on cloud computing.
- Combining pose estimation with other data types such as audio or environmental sensors to create more comprehensive models for applications like smart home, system and healthcare monitoring.
- Utilizing advancements in deep learning such as transformer models for more effective pose estimation.

- Applying pose estimation for real-time behavioral analysis in various settings from retail to healthcare to drive actionable insights.

5.2 Conclusion:

In conclusion, Human Pose Estimation using machine learning represents a rapidly evolving field with vast potential across diverse applications from enhancing fitness and rehabilitation to enabling immersive experiences in augmented and virtual reality the implications of accurate pose estimation are profound.

As advancements in algorithms and computing power, we can expect improvements in accuracy, real-time processing capabilities, and integration with other modalities, leading to more intuitive and responsive systems.

Our analysis found that CNN and RNN are the most common type of deep learning used in 2D human pose estimation. CNN works well in detecting human body joints from a single image. Additionally, the approaches that hardly maintain the performance between accuracy and efficiency must be improved. Since the occlusion and crowded scenario are still the main challenges in human pose estimation.

Human pose estimation is the aim to predict the location of human joints from images and videos. This task is used in many applications, such as.

- Sports Analysis
- Robotics
- Security Surveillance System.
- AI fitness Tracker
- Healthcare Monitoring System
- AR/VR
- Application Interactio

REFERENCES

- I. Jain, A., Tompson, J., LeCun, Y., & Bregler, C. (2015). Modeep: A deep learning framework using motion features for human pose estimation. In *Computer Vision--ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12* (pp. 302-315). Springer International Publishing.
- II. Gupta, A., Gupta, K., Gupta, K., & Gupta, K. (2021). Human Activity Recognition Using Pose Estimation and Machine Learning Algorithm. In *ISIC* (Vol. 21, pp. 25-27).
- III. Arunnehr, J., Nandhana Davi, A. K., Sharan, R. R., & Nambiar, P. G. (2020). Human pose estimation and activity classification using machine learning approach. In *Soft Computing and Signal Processing: Proceedings of 2nd ICSCSP 2019 2* (pp. 113-123). Springer Singapore.
- IV. Jain, A., Tompson, J., Andriluka, M., Taylor, G. W., & Bregler, C. (2013). Learning human pose estimation features with convolutional networks. *arXiv preprint arXiv:1312.7302*.
- V. Luvizon, D. C., Picard, D., & Tabia, H. (2020). Multi-task deep learning for real-time 3D human pose estimation and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(8), 2752-2764.
- VI. Luvizon, D. C., Picard, D., & Tabia, H. (2020). Multi-task deep learning for real-time 3D human pose estimation and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(8), 2752-2764.
- VII. Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., ... & Shah, M. (2023). Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1), 1-37.
- VIII. Luo, H., Wang, M., Wong, P. K. Y., & Cheng, J. C. (2020). Full body pose estimation of construction equipment using computer vision and deep learning techniques. *Automation in construction*, 110, 103016.
- IX. Zimmermann, C., Welschehold, T., Dornhege, C., Burgard, W., & Brox, T. (2018, May). 3d human pose estimation in rgb-d images for robotic task learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1986-1992). IEEE.
- X. Zhang, F., Zhu, X., & Ye, M. (2019). Fast human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3517-3526).
- XI. Park, S., Hwang, J., & Kwak, N. (2016). 3D human pose estimation using convolutional neural networks with 2D pose information. In *Computer Vision--ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14* (pp. 156-169). Springer International Publishing.
- XII. Fieraru, M., Khoreva, A., Pishchulin, L., & Schiele, B. (2018). Learning to refine human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 205-214).
- XIII. Nishani, E., & Çiço, B. (2017, June). Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation. In *2017 6th Mediterranean Conference on Embedded Computing (MECO)* (pp. 1-4). IEEE.